Parmi les problèmes qu'on va résoudre :

Problème n°1 :

Un algorithme qui à partir d'une date donnée non bissextile (le mois de février comptera 28 jours) calcule
et affiche la date du lendemain.
Voici quelques exemples :
Si jour=1, mois=12 et année=1930 **alors** la date du lendemain aura pour jour:2, mois:12 et année= 1930.
Si jour=30, mois=4 et année= 2017 **alors** la date du lendemain aura pour jour:1, mois:5 et année= 2017.
Si jour=31, mois=12 et année= 2018 **alors** la date du lendemain aura pour jour:1, mois:1 et année= 2019.

```
int main() {
      int j, m, a;
      printf("Donner le jour\n");
      scanf("%d", &j);
      printf("Donner le mois\n");  //4 6 9 11
      scanf("%d", &m);
      printf("Donner l\'ann\ée\n");
      scanf("%d", &a);

      if (m == 4 || m == 6 || m == 9 || m == 11) // mois contient 30 jours
                  {
            if (j < 30)
                  printf("le lendemain est %d/%d/%d\n", j + 1, m, a);
            else
                  printf("Le lendemain est %d/%d/%d\n", 1, m + 1, a);
      } else if (m == 2) // mois contient 28 jours
                  {
            if (j < 28)
                  printf("le lendemain est %d/%d/%d\n", j + 1, m, a);
            else
                  printf("Le lendemain est %d/%d/%d\n", 1, m + 1, a);
      } else  //mois contient 31 jours
      if (j < 31)
            printf("le lendemain est %d/%d/%d\n", j + 1, m, a);
      else {
            if (m < 12)
                  printf("Le lendemain est %d/%d/%d\n", 1, m + 1, a);
            else
                  printf("Le lendemain est %d/%d/%d\n", 1, 1, a + 1);
      }
      return 0;
}
```

Problème n°2 :

Un algorithme qui 1) lit deux entiers et vérifie qu'ils sont strictement positifs et inférieurs à 100, 2) calcule et affiche le produit de ces deux entiers par la méthode « **multiplicationRusse** ».

Explication : La méthode « **multiplication Russe** » permet la multiplication de deux entiers **A** et **B** en utilisant seulement la **multiplication** par **2**, la **division** par **2** et l'**addition**.
Exemple : pour **A =17** et **B = 19**, le produit de A par B se fait comme suit :

A B

17	19

Le nombre A est divisé par **2** (**division entière**) et **le nombre B** est **multiplié** par **2,** ce qui donne :

A	**B**
8	38

Le processus se répète jusqu'à avoir dans la **1ère colonne** la valeur **1**.
Le **résultat** *(c'est-à-dire le produit de A par B)* est la **somme des nombres** de la **2ème colonne** qui sont en face des nombres **impairs** de la **1ère** colonne (*Les nombres de la deuxième colonne qui sont en face des nombres pairs de la première colonne sont ignorés*).

	1ère colonne	**2ème colonne**	
→	17	**19**	
	8	38	(*38 est ignoré dans la somme car 8 est paire*)
	4	76	(*76 est ignoré dans la somme car 4 est paire*)
	2	152	(*152 est ignoré dans la somme car 2 il est paire*
→	1	**304**	(*304 fait partie de la somme car 1 est impaire*)

Ce qui donne : **17 * 19 = 19 + 304 = 323**

```
int main() {
      int A, B, A1, B1, mr;
      //répéter lire(A) jusqu' à (A>0 && A <100) ;
      //do ... while(vrai);
      do { // lecture de A
            printf("Donner la valeur de A\n");
            scanf("%d", &A);
      } while (A <= 0 || A > 100);

      do { // lecture de B
            printf("Donner la valeur de B\n");
            scanf("%d", &B);
      } while (B < 0 || B > 100);
      A1 = A;
      B1 = B;
      mr = 0;
      do {
            if (A1 % 2 == 1)
                  mr = mr + B1;
            printf("A = %d  B  = %d  mr = %d\n", A1, B1, mr);
            A1 = A1 / 2;
            B1 = B1 * 2;
      } while (A1 >= 1);
      printf("\n%d * %d = %d", A, B, mr);
      return 0;
}
```

Problème n°3 :

Un algorithme qui affiche tous les nombres inférieurs à 1000 égaux à la somme des cubes de leurs chiffres.

Exemple : $153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27$

Remarque : On divise un nombre pas 10 pour obtenir les chiffres qui le composent

```
int main() {
      int n, n1, chiffre, s;
      for (n = 1; n < 1000; n++) {
            s = 0;
            n1 = n;    //0 =
```

```
            do {
                  chiffre = n1 % 10;
                  s = s + pow(chiffre, 3);
                  n1 = n1 / 10;
            } while (n1 > 0);
            if (n == s)
                  printf("%d\n", n);
      }

      return 0;
}
```

Problème n°4 :

Calcule de $S = 1 - \frac{1}{5*1!} + \frac{1}{25*2!} - \frac{1}{125*3!} + \cdots \pm \frac{1}{5^n*n!}$ $\quad n > 0$

```
int main() {
      int i, n, f, puiss;
      float s, temp;
      do  // lecture de n, n doit être positif
      {
            printf("Donner un nombre positif\n");
            scanf("%d", &n);
      } while (n <= 0);
      s = 1;
      printf("s = %.0f", s);
      puiss = 1;
      f = 1;
      for (i = 1; i <= n; i++) {
            puiss = puiss * -5; //calcule de 5^i
            f = f * i; //calcule de i!
            temp = (float) 1 / (puiss * f); // Calcule  de 1/(-5^i * i!)
            s = s + temp; //s = s+ 1/(-5^i * i!)
            printf(" + (1/(%d*%d)", puiss, f);
      }
      printf("\n");
      printf("s = %f\n", s);
      return 0;
}
```